The D3 Middleware Architecture

Joan Walton, Robert E. Filman*, David J. Korsmeyer, Diana D. Lee†, Ron Mak*, and Tarang Patel†
NASA Ames Research Center, MS 269-2
Moffett Field, CA 94035
* Research Institute for Advanced Computer Science
† Science Applications International Corporation

DARWIN is a NASA developed, Internet-based system for enabling aerospace researchers to securely and remotely access and collaborate on the analysis of aerospace vehicle design data, primarily the results of wind-tunnel testing and numeric (e.g., computational fluid-dynamics) model executions. DARWIN captures, stores and indexes data; manages derived knowledge (such as visualizations across multiple datasets); and provides an environment for designers to collaborate in the analysis of test results. DARWIN is an interesting application because it supports high-volumes of data, integrates multiple modalities of data display (e.g., images and data visualizations), and provides non-trivial access control mechanisms. DARWIN enables collaboration by allowing not only sharing visualizations of data, but also commentary about and views of data.

Here we provide an overview of the architecture of D3, the third generation of DARWIN. Earlier versions of DARWIN were characterized by browser-based interfaces and a hodge-podge of server technologies: CGI scripts, applets, PERL, and so forth. But browsers proved difficult to control, and a proliferation of computational mechanisms proved inefficient and difficult to maintain. D3 substitutes a pure-Java approach for that medley: A Java client communicates (though RMI over HTTPS) with a Java-based application server. Code on the server accesses information from JDBC databases, distributed LDAP security services, and a collaborative information system. D3 is a three tier-architecture, but unlike “E-commerce” applications, the data usage pattern suggests different strategies than traditional Enterprise Java Beans—we need to move volumes of related data together, considerable processing happens on the client, and the “business logic” on the server-side is primarily data integration and collaboration. With D3, we are extending DARWIN to handle other data domains and to be a distributed system, where a single login allows a user transparent access to test results from multiple servers and authority domains.